

SECTION II—CLAIMS

Amendment to the Claims:

This listing of the claims will replace all prior versions and listings of claims in the application. Claim 40, 45, and 50 are amended herein. Claims 1-39 remain canceled herein without prejudice. No new claims are added.

Listing of Claims:

1-39. (Cancelled).

40. (Currently amended) A method for modifying an application to provide functionality for tracing a program flow of the application at a user-configurable level of granularity specified via a Graphical User Interface presented at an end-user device, the method comprising:

reading program code from memory and processing said program code with one or more

processors to perform the following:

presenting to the end-user device via the Graphical User Interface, providing a user with options for modifying the an application's bytecode by injecting tracing and debugging operations into the application's bytecode at the user-configurable level of granularity specified via the Graphical User Interface, wherein the application is, said application composed of a plurality of archive files associated with two or more distinct tiers in a multi-tiered architecture, the said archive files having respective class files, and the said respective class files having respective methods, and wherein the said options for modifying the application's bytecode includes:

~~including one or more of the following:~~ i)

modifying bytecode of only ~~one classfile~~ a subset of a plurality methods from which the application is composed, the subset of the plurality of methods selected from two or more class files in two or more within any one of said archive files composing the application's bytecode as specified via the Graphical User Interface presented to the end-user device, wherein the two or more archive files are associated with two or more distinct tiers of the multi-tiered architecture, and wherein the modified subset of the plurality of methods specified provides the user-configurable level of granularity by providing the functionality for tracing the program flow of the application through only the subset of the plurality of methods specified via the tracing and debugging operations injected into the subset of the plurality of methods specified;

ii) modifying bytecode of only one method within only one of said archive files' respective classfiles;

modifying bytecode of said application in accordance with said user's selection of one of said options;

executing the said application in an object oriented runtime frame work, wherein said executing the application includes including processing a portion of the said application's bytecode that was modified in accordance with the user-configurable level of granularity specified via the Graphical User Interface; and said user's selection of one or more of said options; and,

presenting to said user an output generated from execution of the said portion of the said application's bytecode that was modified to the end-user device via the Graphical User

Interface.

41. (Previously Presented) The method of claim 40 wherein said object oriented runtime framework is a Java compliant object oriented runtime framework.
42. (Previously Presented) The method of claim 40 wherein said portion of said application's bytecode that was modified includes a method entry or method exit.
43. (Previously Presented) The method of claim 42 wherein said output is provided by a plug-in that a bytecode instruction inserted at said method entry or method exit is dispatched to.
44. (Previously Presented) The method of claim 43 wherein said output includes one of:
 - a time at which said method entry or method exit was entered; and,
 - a parameter that is passed at said method entry or method exit.
45. (Currently amended) A machine readable storage medium containing program code that when processed by one or more processors of a computer causes a method to be performed:

presenting to an end-user device via a Graphical User Interface, options for modifying the application's bytecode by injecting tracing and debugging operations into the application's bytecode at a user-configurable level of granularity specified via the Graphical User Interface, wherein the application is composed of a plurality of archive files associated with two or more distinct tiers in a multi-tiered architecture, the archive files having respective having respective methods therein, and wherein the options for modifying the application's bytecode includes:

modifying bytecode of only a subset of a plurality methods from which the application is composed, the subset of the plurality of methods selected from two or more archive files composing the application's bytecode as specified via the Graphical User Interface presented to the end-user device, wherein the two or more archive

files are associated with two or more distinct tiers of the multi-tiered architecture,
and wherein the modified subset of the plurality of methods specified provides the
user-configurable level of granularity by providing the functionality for tracing
the program flow of the application through only the subset of the plurality of
methods;

executing the application in an object oriented runtime frame work; and
presenting output generated from execution of the portion of the application's bytecode that was
modified to the end-user device via the Graphical User Interface.

~~providing a user with options for modifying an application's bytecode, said application~~
~~composed of a plurality of archive files, said archive files having respective class files,~~
~~said respective class files having respective methods, said options including one or more~~
~~of the following:~~

— i) ~~modifying bytecode of only one classfile within any one of said archive files;~~
ii) ~~modifying bytecode of only one method within only one of said archive files' respective~~
~~classfiles;~~

~~modifying bytecode of said application in accordance with said user's selection of one of said~~
~~options;~~

~~executing said application in an object oriented runtime frame work, said executing including~~
~~processing a portion of said application's bytecode that was modified in accordance with~~
~~said user's selection of one or more of said options; and,~~

~~presenting to said user an output generated from execution of said portion of said application's~~
~~bytecode that was modified.~~

46. (Previously Presented) The machine readable storage medium of claim 45 wherein said

object oriented runtime framework is a Java compliant object oriented runtime framework.

47. (Previously Presented) The machine readable storage medium of claim 45 wherein said portion of said application's bytecode that was modified includes a method entry or method exit.
48. (Previously Presented) The machine readable storage medium of claim 47 wherein said output is provided by a plug-in that a bytecode instruction inserted at said method entry or method exit is dispatched to.
49. (Previously Presented) The machine readable storage medium of claim 48 wherein said output includes one of:
 - a time at which said method entry or method exit was entered; and,
 - a parameter that is passed at said method entry or method exit.
50. (Currently amended) A computer comprising ~~containing program code stored in memory of said computer that when processed by one or more processors of said computer causes a method to be performed:~~
a Graphical User Interface to present options for modifying an application's bytecode by injecting tracing and debugging operations into the application's bytecode at a user-configurable level of granularity specified via the Graphical User Interface, wherein the application is composed of a plurality of archive files associated with two or more distinct tiers in a multi-tiered architecture, the archive files having respective methods therein, and wherein the options for modifying the application's bytecode includes: modifying bytecode of only a subset of a plurality of methods from which the application is composed, the subset of the plurality of methods selected from two or more class

files in two or more archive files composing the application's bytecode as
specified via the Graphical User Interface, wherein the two or more archive files
are associated with two or more distinct tiers of the multi-tiered architecture, and
wherein the modified subset of the plurality of methods specified provides the
user-configurable level of granularity by providing the functionality for tracing
the program flow of the application through only the subset of the plurality of
methods specified; and

the Graphical User Interface to further present output generated from execution of the portion of
the application's bytecode that was modified.

~~providing a user with options for modifying an application's bytecode, said application composed of a plurality of archive files, said archive files having respective class files, said respective class files having respective methods, said options including one or more of the following:~~

- i) ~~modifying bytecode of only one classfile within any one of said archive files;~~
- ii) ~~modifying bytecode of only one method within only one of said archive files' respective classfiles;~~

~~modifying bytecode of said application in accordance with said user's selection of one of said options;~~

~~executing said application in an object oriented runtime frame work, said executing including processing a portion of said application's bytecode that was modified in accordance with said user's selection of one or more of said options; and,~~

~~presenting to said user an output generated from execution of said portion of said application's bytecode that was modified.~~

51. (Previously Presented) The computer of claim 50 wherein said object oriented runtime framework is a Java compliant object oriented runtime framework.
52. (Previously Presented) The computer of claim 50 wherein said portion of said application's bytecode that was modified includes a method entry or method exit.
53. (Previously Presented) The computer of claim 52 wherein said output is provided by a plug-in that a bytecode instruction inserted at said method entry or method exit is dispatched to.
54. (Previously Presented) The computer of claim 53 wherein said output includes one of:
a time at which said method entry or method exit was entered; and,
a parameter that is passed at said method entry or method exit.